UAS Conflict-Avoidance Using Multiagent RL with Abstract Strategy Type Communication

Carrie Rebhuhn
Oregon State University
rebhuhnc@engr.oregonstate.edu

Matt Knudson
NASA Ames Research Center
matt.knudson@nasa.gov

Kagan Tumer
Oregon State University
kagan.tumer@oregonstate.edu

Abstract

The use of unmanned aerial systems (UAS) in the national airspace is of growing interest to the research community. Safety and scalability of control algorithms are key to the successful integration of autonomous system into a human-populated airspace. In order to ensure safety while still maintaining efficient paths of travel, these algorithms must also accommodate heterogeneity of path strategies of its neighbors. We show that, using multiagent RL, we can improve the speed with which conflicts are resolved in cases with up to 80 aircraft within a section of the airspace. In addition, we show that the introduction of abstract agent strategy types to partition the state space is helpful in resolving conflicts, particularly in high congestion.

Introduction

The air traffic density in the national airspace (NAS) is increasing beyond the current capabilities of air traffic controllers. Centralized human control at each sector in the airspace works well for low plane-to-controller ratios, but with the growth commercial air traffic and the future introduction of autonomous systems in the airspace, it is clear that control algorithms must begin to handle some of the safety in the airspace. The FAA NextGen Implementation Plan promises regulation mandating that automatic dependent surveillance broadcast (ADS-B) systems be installed on all aircraft, providing a method by which air traffic control systems can easily gather information about sector congestion. More promising from a multiagent standpoint is the introduction of traffic information service broadcast (TIS-B), which provides distributed automatic ADS-B information to planes within a 15-mile radius of other planes. Using this locally-available information, we can construct conflict-avoidance in the national airspace as a distributed multiagent problem.

Q-Learning in the NAS

Conflict-avoidance in the UAS domain requires an agent selects the parameters of a *diversion waypoint* in a way that maintains safety while still considering the path cost of a diversion and the potential for conflict propagation in the system. In this work we use reinforcement learning agents to map plane states (relative to the plane's nearest neighbor) to conflict-avoidance actions (waypoints) through Q-learning. A point-mass simulator is used to model agent trajectories, and agents must make a conflict-avoidance decision when the simulator detects a path conflict in some time horizon.

We call the joint state of all agents $\vec{s}$. Each agent i has a state s_i, which is described in relation to its nearest neighbor, such that $s_i = \{\delta_{n(i)}, \Theta_{n(i)}, h_{n(i)}, \mathcal{T}_{n(i)}, p_i, g_i\}$, where $\delta_{n(i)}$ is the xy-planar distance to the nearest neighbor of i, $\Theta_{n(i)}$ is the relative heading of the nearest neighbor of i, $h_{n(i)}$ is the relative height (z-position) of the nearest neighbor of i, $\mathcal{T}_{n(i)}$ is the type of the nearest neighbor, p_i is the agent's position, and g_i is the goal position of the agent. The position p_i and goal position g_i of the agent is not useful for the task of conflict-avoidance, so for the purpose of Q-learning we do not distinguish between states with different p_i or g_i values. The type information $\mathcal{T}_{n(i)}$ is used to distinguish different states in Q-learning in our *type-partitioned* experiments, but we compare to when this is not used to distinguish types in our *type-free* experiments. Agents select an action $a_i = \{\tau, m, t\}$, where τ is the action type (heading change, or altitude change), m is the magnitude of this change, and t is the duration of the redirection. Agents use $\epsilon-$greedy action selection to choose an action a_i based on their a state s_i, and then receive a reward $R(\vec{s}, \vec{a})$ based on the system state $\vec{s}$ and the action taken by the agent.

This reward captures the cost of conflicts caused after the course correction, the signal cost of taking a diversion action, and the distance cost of a particular action. We formalize our the local reward as:

$$L_i(s_i, a_i) = w_c u(1-n_c(s_i)) - w_a u(n_a(a_i)) - w_d d_{extra}(s_i, a_i) \quad (1)$$

where $L_i(s_i, a_i)$ is the local reward given by the agent's state s_i and the agent's action selection a_i, $n_c(s_i)$ is the number of conflicts that agent i is involved with, $n_a(a_i)$ is the number of deviations agent i takes, $d_{extra}(s_i, a_i)$ is the amount of extra distance that will be added by creating the diversion waypoint, and u is the unit step function. In our setup, the values for these parameters are $w_c = 10, w_d = 0.1, w_a = 10$.

Local rewards perform well in domains without congestion, but when resources cannot be shared equally to attain optimality, global rewards can promote coordination. Our global reward is a simple summation of the local rewards:

$$G(\vec{s}, \vec{a}) = \sum_{i=0}^{N} L_i(s_i, a_i) \tag{2}$$

Because global rewards can sometimes be too noisy for agents to learn, we also test a 'difference' reward, which evaluates an agent based on the effect of removing it from the system. We derive this reward using the global reward and the difference reward equation given in Tumer et al. (Agogino and Tumer 2008):

$$D_j(\vec{s}, \vec{a}) = \sum_{i \in C_j}^{N} L_i(s_i, a_i) \tag{3}$$

where $D_j(\vec{s}, \vec{a})$ is the difference reward for agent j and C_j is the set of agents in projected conflict with j.

Agents in the system have strategy types that define generally what conflict-avoidance action they will take. There are four different strategy types; one that always rerouted CCW by $90°$, one that always rerouted CW by $90°$, one that always changed its altitude by $\pm 1000m$, and one that learned at a rate of $\alpha = 0.1$ and $\gamma = 0.9$ with a random action-selection chance of $\epsilon = 0.1$. To take advantage of this type information, an agent was pre-trained with each of the heuristic types in an environment with 5 agents with targets and initial positions created within a 1000m^3 cube. The Q-table developed by the learning agent, or *stereotype*, was then used to initialize the agents in the test system.

Results and Discussion

We are interested in two things: the performance of reward structures and scalability of reward structures with the inclusion of types in the state space. Figure 1 shows the impact of agent types on performance with high congestion. Types provide good initialization for all reward structures in the high-congestion domain. They also improve the convergence of global and local rewards.

Figure 2 shows that with the difference reward, at low levels of congestion the agents find optimal policies, and there is differentiation between the performance while identifying types and the performance without the inclusion of types. At all levels of congestion the use of types improves the initial performance of the UAS in conflict-avoidance. However this improvement comes at the cost of the learning speed, through partitioning of the state space. In high congestion, learning without types overtakes learning with types, but converges to a similar solution. In cases of medium congestion, learning without types overtakes learning with types, which decays slightly but then converges to similar performance to learning with types.

In this paper we show that discriminating by agent strategy type improves performance in high congestion in the UAS domain for global and local rewards. Under the difference reward, we show that learning with types initially

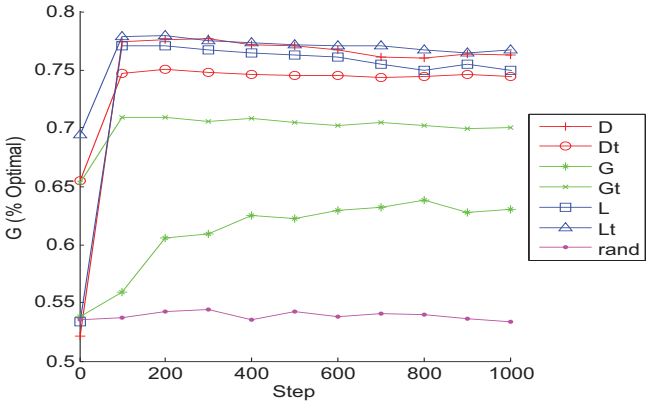

Figure 1: These results show three different learning approaches under high congestion (80 agents), comparing learning with and without types in the state space.

outperforms learning without types, but is quickly overtaken due to a slower learning speed. Values of learning with and without types converge to similar performance under this structure. Observing this, we identify a tradeoff between the usefulness of including neighbor policies in the state space (*policy type value*) and the increase in *training samples* from not separating the Q-table updates by agent type. Knowing type information is valuable in high congestion because it provides necessary information about the local actions of the agents. In cases with lower congestion, however, the *policy type value* is lower because a more general policy is acceptable.

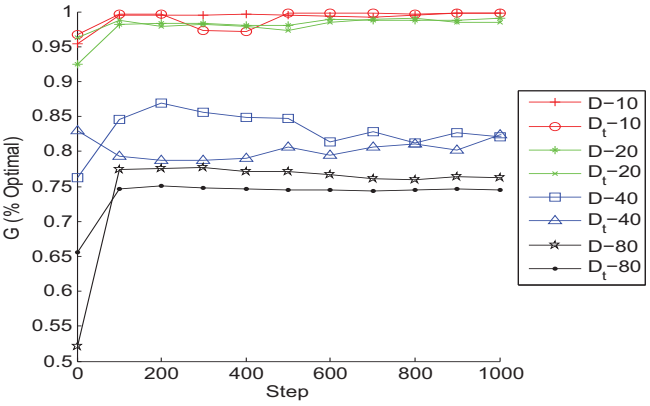

Figure 2: Agent density impact on global performance using difference rewards.

Acknowledgements

This work was completed at NASA AMES Research Center.

References

Agogino, A., and Tumer, K. 2008. Analyzing and visualizing multi-agent rewards in dynamic and stochastic domains. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)* 320–338.